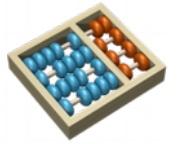




UNIVERSIDADE ESTADUAL DE CAMPINAS
Instituto de Computação – IC
MO611 – Teleprocessamento e Redes
Prof. Dr. Edmundo Roberto Mauro Madeira



THALES EDUARDO NAZATTO

**COMUNICAÇÃO DE DADOS VIA CLOUD E FOG COMPUTING ENVOLVENDO
ELEMENTOS DE CIÊNCIA DE DADOS (APRENDIZADO DE MÁQUINA E
RECONHECIMENTO DE PADRÕES):** Uma Abordagem *Survey* sobre o tema

CAMPINAS
2019/2

THALES EDUARDO NAZATTO

COMUNICAÇÃO DE DADOS VIA CLOUD E FOG COMPUTING ENVOLVENDO
ELEMENTOS DE CIÊNCIA DE DADOS (APRENDIZADO DE MÁQUINA E
RECONHECIMENTO DE PADRÕES): Uma Abordagem *Survey* sobre o tema

Monografia apresentada ao Instituto de
Computação da Universidade Estadual de
Campinas como parte dos requisitos
exigidos para a conclusão da disciplina
MO611 – Teleprocessamento e Redes,
ministrada no segundo semestre de 2019.

Professor: Prof. Dr. Edmundo Roberto Mauro Madeira

CAMPINAS
2019/2

RESUMO

Até o presente momento, *Cloud computing* é um conceito já consolidado no cenário de tecnologia atual e, com a popularização da *Internet of Things*, o crescimento de dispositivos pequenos capazes de trafegar dados na Internet faz crescer também a *Fog computing*, um conceito atuante na borda da rede capaz de aproveitar a robustez da *Cloud computing* ao mesmo tempo que mantém a agilidade necessária aos dispositivos compatíveis com IoT. Esta monografia é um levantamento (*survey*) sobre como se relacionam os temas ciência de dados e *Cloud/Fog computing* em uma aplicação disponibilizada pela rede, enquanto serão feitas comparações para determinar tendências entre os artigos selecionados.

Palavras-chave: Nuvem, Névoa, Borda da rede, *Cloud computing*, *Fog computing*, *Edge computing*, Dados, Ciência de dados, Aprendizado de máquina, Padrões, Reconhecimento de padrões.

ABSTRACT

At the present moment, *Cloud computing* is a concept that is well established in today's technology landscape, and with the popularization of the *Internet of Things*, the growth of small devices capable to traffic data on the Internet also grows the *Fog computing*, a concept acting on network's edge capable to take advantage of *Cloud computing*'s robustness while maintaining the agility required to IoT compatible devices. This monograph is a survey about how the themes data science and *Cloud/Fog computing* relates in an application published into network, while comparisons will be made to determine trends among selected articles.

Keywords: Cloud, Fog, Network edge, *Cloud computing*, *Fog computing*, *Edge computing*, Data, Data science, Machine learning, Pattern, Pattern recognition.

SUMÁRIO

1 INTRODUÇÃO.....	6
2 CLOUD, FOG E EDGE COMPUTING.....	7
2.1 CLOUD COMPUTING.....	7
2.2 EDGE COMPUTING.....	10
2.3 FOG COMPUTING.....	11
2.4 PROTOCOLOS UTILIZADOS.....	13
3 CIÊNCIA DE DADOS E SEUS ELEMENTOS.....	17
3.1 APRENDIZADO DE MÁQUINA (MACHINE LEARNING).....	17
3.2 RECONHECIMENTO DE PADRÕES.....	19
3.3 APLICAÇÕES NO CENÁRIO DE REDES DE COMPUTADORES.....	21
4 TRABALHOS COLETADOS E COMPARAÇÕES.....	23
5 CONCLUSÃO.....	26
6 BIBLIOGRAFIA.....	27

1 INTRODUÇÃO

Com o seu surgimento, aos poucos a *Cloud computing* ganhou espaço no desenvolvimento de novas aplicações graças a sua praticidade, escalabilidade e redução de custos e atualmente possui plataformas consolidadas como Microsoft Azure e Amazon Web Services. Contudo, o crescimento de dispositivos móveis e sensores com capacidade de acessar a Internet, definidos dentro do grupo da *Internet of Things*, aumenta também as pesquisas relacionadas a serviços que se situam na borda da rede, definidos como *Edge computing* ou *Fog computing* em situações mais específicas.

Ao mesmo tempo que esses serviços cresceram, a crescente capacidade de processamento fez com que aplicações relativas a ciência de dados, como *big data* e *machine learning*, fossem cada vez mais viáveis no cenário atual. Tais aplicações permitem um grande aproveitamento dos dados para análise de padrões e comportamento, podendo gerar um grande valor para uma organização se tais dados estiverem coerentes e o método de análise adequado para tal.

Esta monografia tem como objetivo fazer um levantamento (*survey*) de artigos relativos a *Cloud*, *Edge* e *Fog computing* envolvendo aplicações de ciência de dados, sendo neste caso *machine learning* e reconhecimento de padrões, para demonstrar suas abordagens e viabilidade, comparando informações entre os trabalhos para determinar tendências sobre o casamento entre os dois temas, divididas em cinco capítulos contando com esta introdução. O segundo e terceiro capítulos trazem conteúdos introdutórios sobre *Cloud* e *Fog computing* e ciência de dados. O quarto capítulo traz o levantamento com os artigos encontrados e comparações entre os mesmos através de sua discussão e uma tabela comparativa. Por fim, o quinto e último capítulo traz a conclusão de todo o levantamento.

2 CLOUD, FOG E EDGE COMPUTING

Antes de detalhar sobre os trabalhos pesquisados e comparações sobre eles, primeiro é preciso detalhar em caráter introdutório quais os conceitos que serão abordados nesta monografia, para entendimento das análises abordadas posteriormente. Neste capítulo, serão abordados os assuntos *cloud*, *fog* e *edge computing*, sua colocação no cenário atual e as principais diferenças sobre estes 3 conceitos no cenário de redes de computadores.

2.1 CLOUD COMPUTING

O termo *cloud computing* foi o primeiro dos três a ser cunhado e colocado em prática nas redes mundiais de computadores. O uso de seu conceito possui grande relevância nas aplicações atuais, permitindo com que as aplicações fossem modularizadas entre partes destinadas a processamento bruto de informações (comumente chamada de *backend*) e partes destinadas a interface gráfica e experiência de usuário (comumente chamada de *frontend*), sem que houvessem gastos exorbitantes em infra-estrutura, substituindo o sistema desenvolvido de forma monolítica sem maiores preocupações.

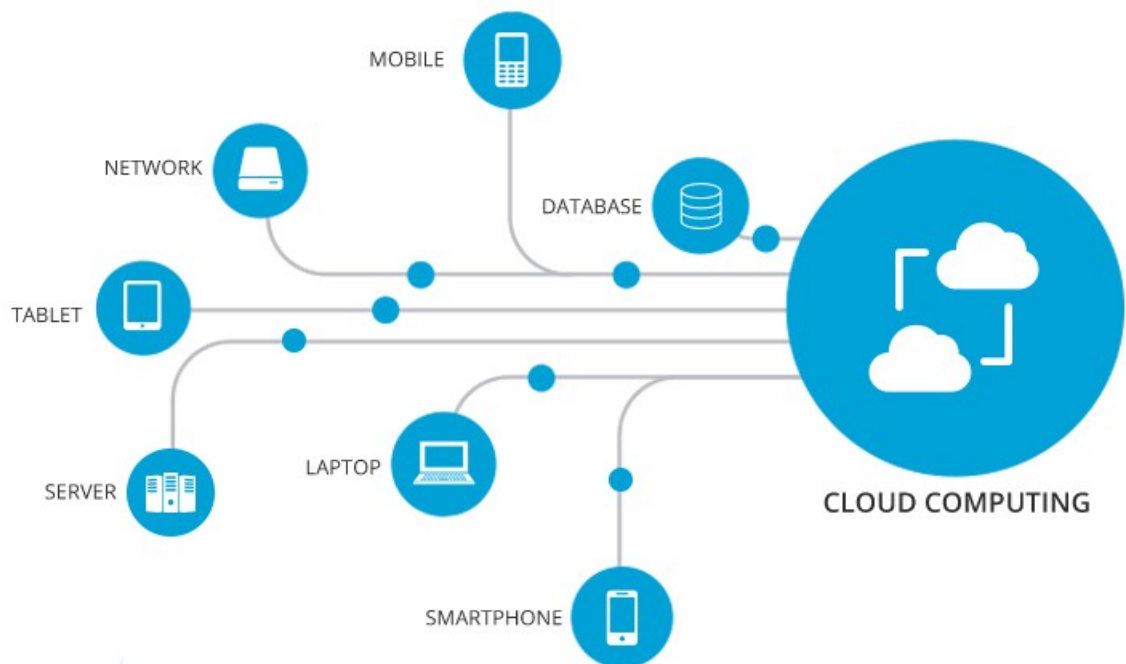


Figura 1. Ilustração sobre o significado de *cloud computing*: O acesso dos dados como serviço na rede.

Este termo tem sua origem em 1996 em um plano de negócios da Compaq. O documento, intitulado “Internet Solutions Division Strategy for Cloud Computing” e escrito por George Favaloro, possuía 50 páginas e detalhava que o software corporativo daria lugar a serviços presentes via Web e que, no futuro, "o software de aplicativo não é mais um recurso de hardware - mas da Internet" [1]. Em 2006, a Amazon lança sua plataforma, o Amazon Web Services [4]; Eric Schmidt, então CEO do Google, introduz o termo em seu contexto moderno em uma conferência realizada em Agosto [1] e, no mesmo mês, a Amazon introduz o EC2, que popularizaria o AWS e o termo *Cloud computing* em todo o mundo [3].

A definição de *Cloud computing* é “um modelo de disponibilização e utilização de Tecnologias de Informação e Comunicação, que permite o acesso remoto, através da internet, a um leque de recursos de computação compartilhados em forma de serviços” [2]. Nela, os recursos podem ser acessados sob demanda não pelo meio físico, mas pela “nuvem”, através de *datacenters* espalhados pelo mundo todo, dependendo da escalabilidade da companhia que está oferecendo o serviço. Dessa forma, em vez de construir e manter uma infra-estrutura de redes complexa e cara, o gasto é mensurável pela quantidade de dados que a aplicação desenvolvida consome, fazendo com que um projeto seja mais barato e tenha mais previsibilidade em seus gastos. A Figura 1 ilustra esse conceito, onde vários tipos de dispositivos estão todos interligados de certa forma requisitando serviços para a “nuvem”, se comunicando das mais variadas formas sem a necessidade de se “conhecerem” diretamente, apenas através dos dados e da “nuvem”.

Quando nos referimos a “nuvem”, nos referimos a própria Internet. O termo “nuvem” (Ou *cloud*, em inglês) já era usado por redes de computadores predecessoras da Internet como a ARPANET em 1977 e a CSNET em 1981, como forma de representar redes em seus diagramas. O motivo desta representação era usar a nuvem como uma metáfora com o intuito de que o tamanho da rede e suas conexões fosse insignificante para o diagrama – e, consequentemente, o projeto – de sua própria rede [3]. Aplicando a metáfora da nuvem nos dias atuais, é possível ver como ela se encaixa perfeitamente: Independentemente de como os serviços estão organizados na companhia em que um cliente solicitará determinado serviço na rede, as únicas preocupações que o cliente deve ter ao realizar esse serviço é se ele será realizado ou não e com qualidade e eficiência. O desenho da rede é apenas uma consequência das preocupações analisadas por seu cliente, podendo migrar para outro provedor de serviços caso não alcancem os resultados necessários.

Como modelos de serviços proporcionados por uma provedora de serviços de *cloud computing*, o NIST (National Institute of Standards and Technology) determinou três modelos de serviço padrão [5], sendo ilustradas na Figura 2. Conforme mencionado anteriormente, em nenhum dos três modelos gerencia ou controla a infraestrutura de nuvem subjacente:

- **Infrastructure as a service (IaaS):** A capacidade fornecida ao cliente é para fornecer processamento, armazenamento, redes e outros recursos de computação fundamentais, nos quais o consumidor pode implantar e executar software arbitrário, que pode incluir sistemas operacionais e aplicativos. Ele tem controle sobre sistemas operacionais, armazenamento e aplicativos implantados; e pode ter controle limitado de componentes de rede selecionados (Ex: firewalls).
- **Platform as a service (PaaS):** A capacidade fornecida ao cliente é para implantar na infraestrutura de nuvem os aplicativos criados ou adquiridos pelo cliente, criados usando linguagens de programação, bibliotecas, serviços e ferramentas suportadas pelo provedor. Ele tem controle sobre os aplicativos implantados e, possivelmente, as definições de configuração para o ambiente de hospedagem de aplicativos.
- **Software as a service (SaaS):** A capacidade fornecida ao cliente é para usar as aplicações do provedor em execução em uma infraestrutura de nuvem. Os aplicativos são acessíveis a partir de vários dispositivos clientes através de uma interface como um navegador da web (Ex: Cliente de e-mail) ou uma interface de programa. Ele pode ter definições limitadas de configuração de aplicativos específicos do usuário.

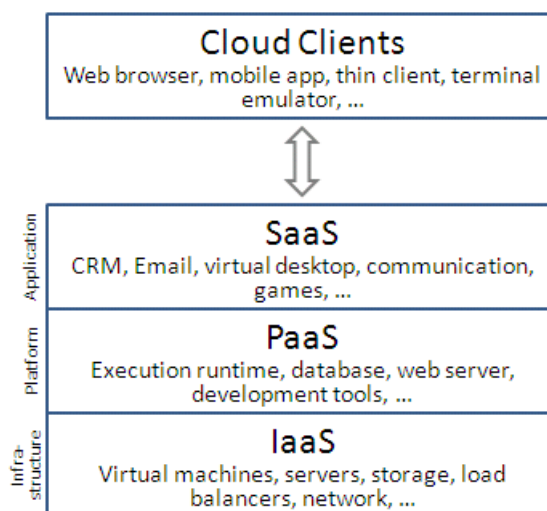


Figura 2. Camadas dos modelos de serviço presentes em um serviço focado em *cloud computing*.

2.2 EDGE COMPUTING

Apesar de revolucionar o desenvolvimento de novas aplicações, um dos grandes problemas do *cloud computing* reside no fato que os *datacenters* de seus provedores de serviço geralmente estão distantes de seus clientes. Por consequência, a latência é um dos principais desafios a serem contornados no processo de desenvolvimento, tendo como alternativas processamento paralelizado ou filas de mensagens para processamento em lote. Entretanto, tais soluções aliviam o gargalo apenas na aplicação, e a topologia da rede permanece inalterada em teoria.

Com o avanço dos dispositivos compatíveis com IoT (Internet das Coisas, ou *Internet of Things*), a latência no envio dos dados passa a ter uma importância maior. Como tais dispositivos costumam enviar dados constantemente, os centros de processamento e as redes que conectam esses centros aos dispositivos podem se tornar gargalos caso esses elementos não sejam projetados com capacidade adequada, podendo levar a aumentos significativos nos níveis de congestionamento na rede de *backbone*, com consequentes atrasos e perda de pacotes, prejudicando a experiência do usuário [7]. Nesse contexto, surge a principal motivação para o desenvolvimento de soluções em *Edge computing*.

O termo *Edge computing* tem sua origem no final dos anos 90, com aplicações desenvolvidas nos laboratórios da Akamai que reduziram cargas de dados colocando *Web caches* nas bordas de sua rede (Dilley et al., 2002). Trata-se de um novo modelo de computação que permite armazenar e processar dados na borda da rede e fornecer serviços inteligentes perto da origem dos dados, colaborando com a *Cloud computing* [9].

A *Edge computing*, conforme ilustrada na Figura 3, não veio para substituir a *Cloud computing*, e sim para complementá-la. Ela geralmente lida com o processamento de dados onde os dados são criados em torno da rede, em substituição ao *datacenter* centralizado de processamento de dados. Para o ponto de entrada, são usados dispositivos na borda que permitem a entrada nas redes principais, podendo ser utilizados para *cache* e processamentos simples executados em tempo real. A computação é realizada em grande parte ou completamente em nós de dispositivos distribuídos, conhecidos como *smart devices* ou *edge devices*, em vez de ocorrer principalmente em um *datacenter* localizado a quilômetros de distância [6]. O *datacenter* na nuvem só seria requisitado de fato em situações onde demandam um grande processamento/armazenamento de dados, tais como processamentos de lógicas de negócio, grandes bancos de dados, análise de um lote de dados, entre outros.

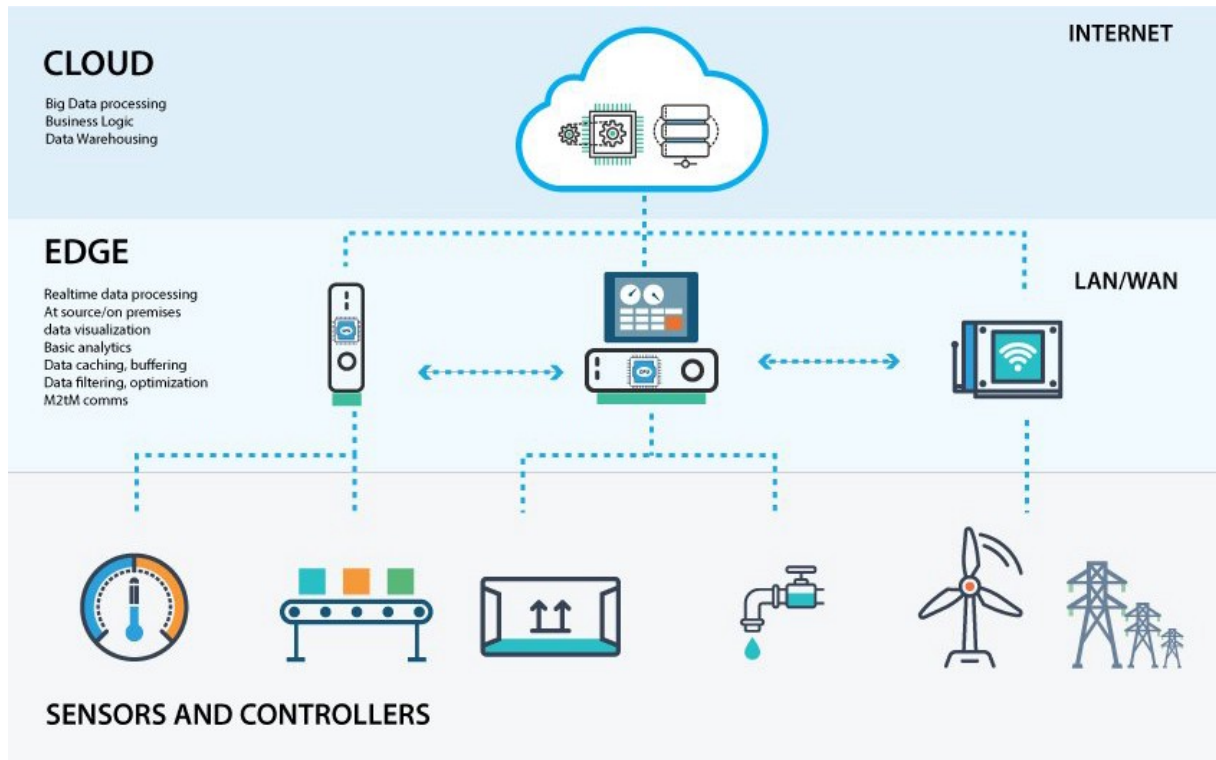


Figura 3. Combinação entre *edge computing* e *cloud computing*, detalhando suas funções.

2.3 FOG COMPUTING

Segundo [10], a *Edge computing* é um modelo que abrange uma série de outros modelos mais específicos, tais como os modelos de *cloudlets* (pequenos *datacenters* localizados a borda da rede), modelos de *mobile edge computing* (capaz de prover serviços de *Cloud computing* em redes móveis), e outros como ITS-Clouds (sistemas de transportes inteligentes em nuvem), VANETs (Redes Ad-Hoc de redes veiculares) e CDN (*Content Delivery Networks*). A *Fog computing* também é percebida como um desses modelos, e também possui suas peculiaridades.

O termo *Fog computing* foi cunhado nos laboratórios da Cisco em 2011 e, assim como a *Cloud computing*, tem sua origem graças a uma metáfora: Assim como a neblina está mais próxima do solo do que a nuvem, a *Fog computing* está mais próxima a borda da rede [9]. Entretanto, o que especifica *Fog computing* em relação a *Edge computing* é que, enquanto a *Edge computing* pode envolver ou não os serviços de *Cloud computing*, *Fog computing* é uma forma de trazer tais serviços a borda da rede.

De acordo com [9], as características definidoras da *Fog computing* são:

- Baixa latência e reconhecimento do local.
- Distribuição geográfica generalizada.

- Mobilidade.
- Número muito grande de nós.
- Papel predominante do acesso sem fio.
- Forte presença de aplicações de *streaming* e em tempo real.
- Heterogeneidade.

Na busca de padronização, em 19 de novembro de 2015 as empresas/instituições Cisco, Dell, Intel, Microsoft, ARM e Princeton University lançaram o *OpenFog Consortium* para desenvolver uma arquitetura de referência aberta. Em 13 de fevereiro de 2017, o *OpenFog Consortium* lança a primeira versão de sua arquitetura de referência.

A arquitetura de *Fog computing* possui múltiplas camadas, ilustradas na Figura 4 e utilizando-se o conceito de *Fog-to-Cloud* (F2C). No geral, o servidor presente na névoa pode ser hospedado em dispositivos das camadas de acesso (borda da rede) ou em camadas mais próximas ao dispositivo final. A *Fog computing* funciona como uma extensão aos serviços da nuvem e como provedor de recursos para os dispositivos de IoT.

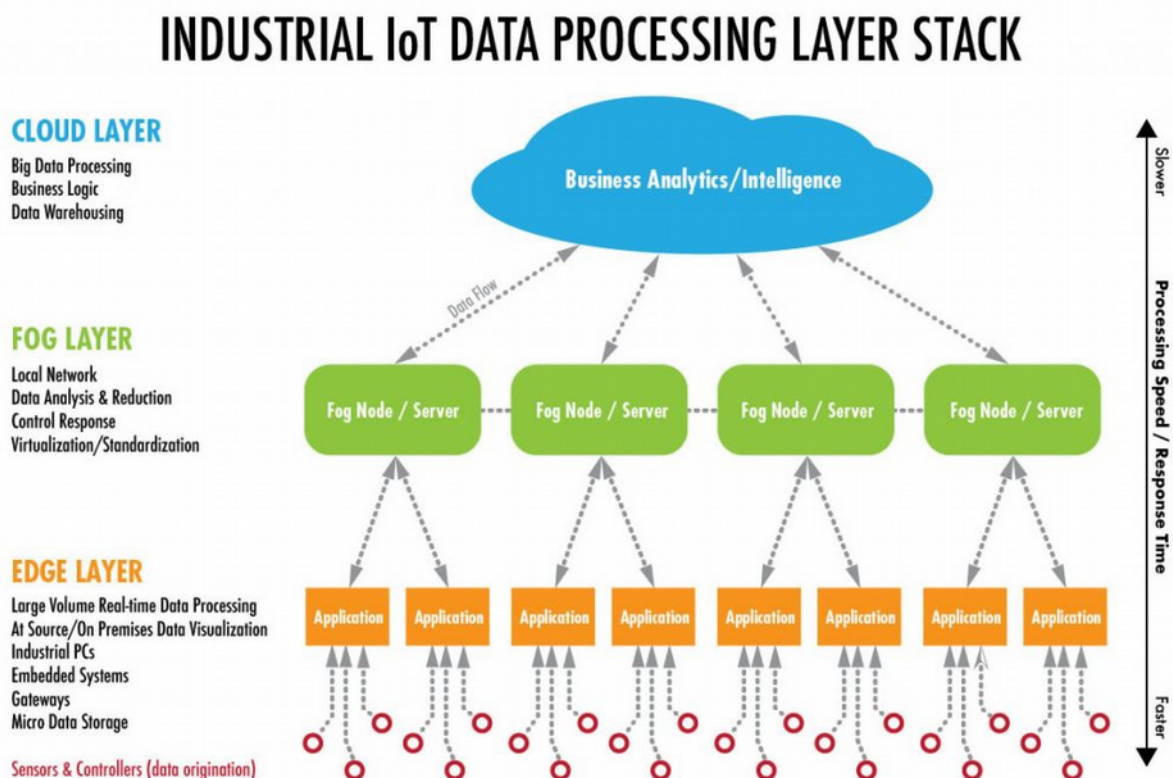


Figura 4. Camadas de uma arquitetura de *Fog computing*.

O conceito de F2C pode ser explorado como base para a definição da arquitetura do sistema, considerando a *Fog computing* como um paradigma complementar à *Cloud*

computing. O principal objetivo é prover a capacidade de executar serviços enquanto garante-se baixo tempo de resposta, redução de carga na rede e melhor eficiência energética. O conceito de F2C baseia-se em uma arquitetura hierárquica, onde cada camada possui sua própria função [12].

Uma arquitetura F2C é baseada em dois domínios de gestão: camadas e áreas. A camada é um conjunto de dispositivos com características e recursos similares; já áreas são conjuntos de dispositivos conectados, próximos fisicamente ou logicamente, em uma mesma camada. Os nós da névoa são os componentes das áreas, enquanto as camadas são representações distribuídas de acordo com suas funcionalidades.

Na Figura 5, está ilustrado a arquitetura para *Fog computing* baseada no conceito de F2C definida pelo *OpenFog Consortium*, presente em [13]. A arquitetura de referência propõe algumas perspectivas (segurança da informação, gerenciamento de recursos, desempenho e escala, controle e análise de dados, negócios e aplicações na névoa), além de recursos e serviços (redes, computação, virtualização, etc).

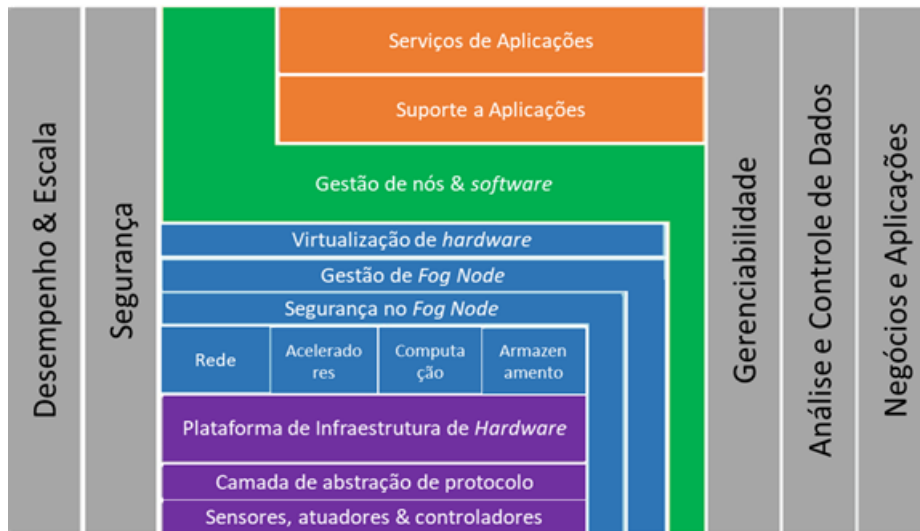


Figura 5. Arquitetura definida pelo *OpenFog Consortium* baseada em perspectivas, recursos e serviços.

2.4 PROTOCOLOS UTILIZADOS

Cloud, *Edge* e *Fog computing* são conceitos utilizados para descrever serviços de computação providos na rede, e protocolos de comunicação são criados ou adaptados para esse fim. Dizdarević et al. faz um levantamento dos principais protocolos utilizados para comunicação em uma arquitetura F2C na camada de aplicação da rede, uma vez que TCP e UDP já são protocolos consolidados na camada de transporte da rede para transporte de dados

confiável e não-confiável, e chegou a um total de 6 protocolos principais: AMQP (*Advanced Message Queuing Protocol*), CoAP (*Constrained Application Protocol*), DDS (*Data Distribution Service*), HTTP (*Hyper Text Transfer Protocol*), MQTT (*Message Queuing Telemetry Transport*) e XMPP (*Extensible Messaging and Presence Protocol*).

Enquanto o HTTP e o CoAP são baseados em modelos de requisição e resposta na arquitetura cliente-servidor, AMQP, DDS, e MQTT possuem um modelo diferente, chamado publicar-subscriver e presente na Figura 6. Este modelo consiste em três partes, publicador, assinante e um *broker*, onde o cliente com a função de assinante não precisa solicitar informações do servidor, e sim assinar eventos do *broker*, que atua como provedor de informações, filtro e roteador de mensagens entre publicador e assinante. Este modelo é interessante para *Cloud*, *Edge* e *Fog computing* pois fornece escalabilidade e simplifica interconexões entre dispositivos diferentes, suportando comunicação dinâmica e assíncrona. O XMPP pode assumir tanto o modelo cliente-servidor como o modelo publicar-subscriver, embora em suas primeiras versões adote apenas o modelo cliente-servidor.

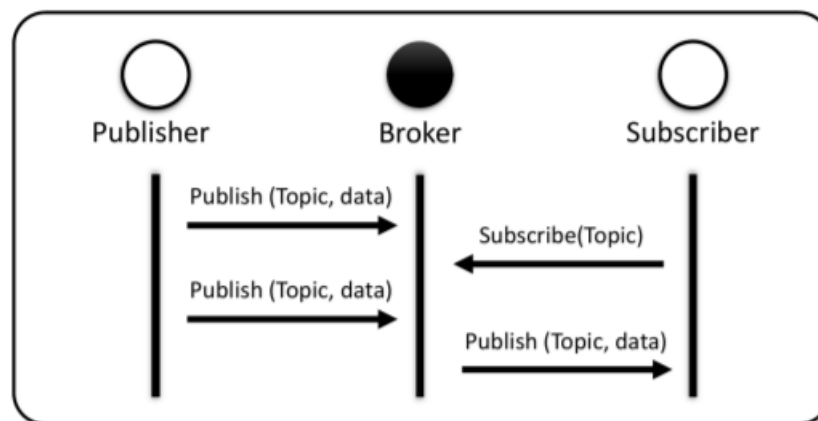


Figura 6. Modelo publicar-subscriver presente nos protocolos AMQP, DDS e MQTT.

O HTTP é comumente associado ao padrão REST (*Representational State Transfer*) para realizar as chamadas nos serviços presentes na nuvem e na névoa. Um exemplo desta interação é ilustrado na Figura 7, onde um cliente faz uma requisição POST para inserir um dado no servidor, que devolve um código de resposta 201 para indicar sucesso, e um outro cliente faz uma requisição GET para receber um dado dentro do servidor, que devolve o dado como resposta e um código de resposta 200 para indicar sucesso. Embora não seja uma obrigação, o padrão REST ajuda muito os desenvolvedores na hora de realizar os mapeamentos para realizar operações de trocas de dados entre nós da rede. Geralmente

requisições e respostas em JSON são utilizadas nos serviços, por ser um padrão mais leve que o XML.



Figura 7. Exemplo de chamadas de 2 clientes utilizando o padrão REST para o protocolo HTTP.

O CoAP usa o mesmo modelo cliente-servidor do HTTP e também se aproveita do padrão REST, mas difere utilizando o UDP ao invés do TCP na camada de transporte da rede, permitindo uma comunicação de dados mais rápida embora não confiável. O CoAP é considerado leve não apenas por utilizar UDP, mas também por encodar todos os cabeçalhos, métodos e códigos de status das mensagens, sendo recomendável para ambientes de processamento limitado e altas restrições.

O MQTT também é recomendável para as mesmas situações do CoAP, porém usa o modelo publicar-subscriver e usa TCP na camada de transporte da rede. Por ser projetado para ser mais leve, possui cabeçalhos de tamanho pequeno em suas mensagens e não possui suporte a encriptação. Em contrapartida, o AMQP é projetado para ser um protocolo robusto, projetado para permitir a interoperabilidade entre uma ampla variedade de aplicativos e sistemas diferentes, independentemente de como foram projetados, mas possui a desvantagem de ser um protocolo de altos requerimentos de processamento e memória. Possui suporte a encriptação via TLS (*Transport Layer Security*) e autenticação via SASL (*Simple Authentication and Security Layer*).

Em contrapartida a esses dois, o DDS é descentralizado e baseado na comunicação *peer-to-peer* e, como tal, não depende do *broker*. No DDS, publicadores e assinantes podem se comunicar como pares por meio do barramento de dados, permitindo a troca de dados assíncrona com base em seus interesses. O fato de não haver intermediário também diminui a probabilidade de falha do sistema, porque não há um ponto único de falha para todo o sistema. Os dois lados da comunicação são dissociados e um publicador pode publicar dados mesmo que não haja assinantes interessados. Como os publicadores não perguntam sobre quem consome seus dados, o uso de dados é fundamentalmente anônimo.

Já o XMPP foi projetado inicialmente como um protocolo para mensagens instantâneas e troca de mensagens entre aplicações, utilizando o formato XML como padrão para formação de suas mensagens. Embora suas mensagens sejam um pouco mais pesadas do

que a média pela escolha do formato, possui características interessantes para encriptação de mensagens, contendo o TLS intrínseco a sua especificação e o SASL para autenticação.

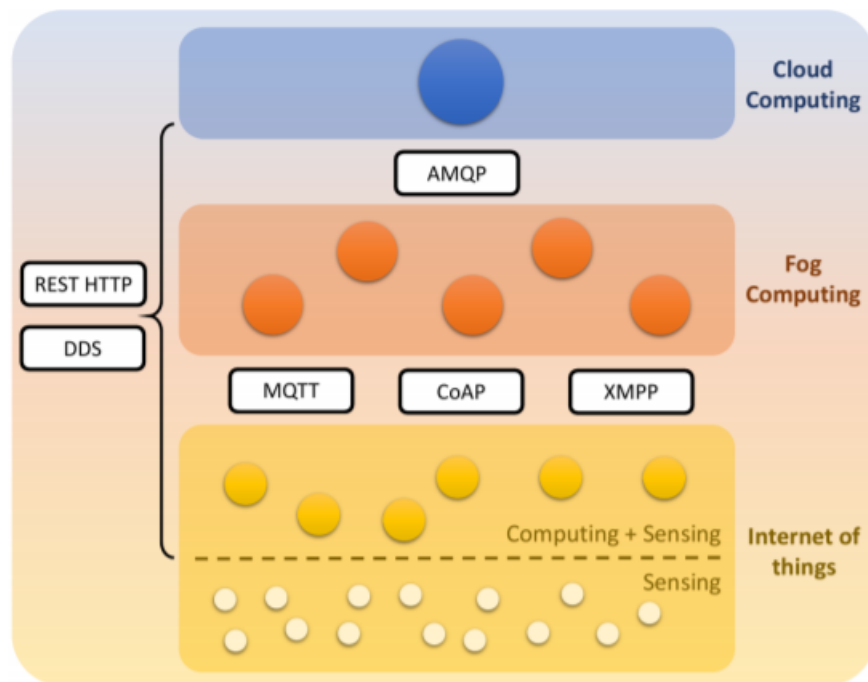


Figura 8. Protocolos mais adequados a cada situação em uma arquitetura F2C

Dada a visão básica destes protocolos, já é possível perceber que alguns deles casam mais com a característica de transportar tais dados para toda a rede enquanto outros são mais propícios a serem intermediários na troca de mensagem entre camadas. Também pode-se perceber que alguns destes protocolos se adequam mais aos requisitos de *Cloud computing* enquanto outros são mais favoráveis a *Edge/Fog computing*. Dizdarević et al. ainda mostra que a versatilidade do HTTP e do DDS permitem que eles sejam utilizados por todo o fluxo do dado na rede, a robustez do AMQP é ideal para estabelecer a comunicação dos servidores presentes na névoa com os *datacenters* da nuvem e protocolos mais leves como o MQTT, CoAP e XMPP podem estabelecer a comunicação entre os dispositivos IoT e os servidores presentes na névoa, por demandarem menos recursos e, dessa forma, garantindo latência menor e entrega constante de dados dos dispositivos IoT.

3 CIÊNCIA DE DADOS E SEUS ELEMENTOS

Ciência de dados (*Data Science*, em inglês) é uma área que engloba conceitos de Computação, Estatística, Engenharia e outras áreas dependendo do dado a ser explorado. Segundo [15], ela é voltada para o estudo e a análise de dados, estruturados ou não, que visa a extração de conhecimento ou *insights* para possíveis tomadas de decisão para atingirem melhor eficiência e resultados, de maneira similar à mineração de dados. Suas aplicações podem atingir os mais variados tipos de negócios, desde sistemas de recomendação em sites de compras, aumento de alcance com propagandas personalizadas, buscas na internet mais eficientes devido ao entendimento do contexto usado pela pessoa ao usar o buscador, análise de crédito ou uso de reconhecimento de imagens para detectar padrões.

A ciência de dados existe há cerca de 30 anos, porém se popularizou nos últimos anos com o surgimento do *Big Data*, a maior capacidade de processamento atingida recentemente e o desenvolvimento de áreas como o *Machine Learning*. Não a toa, essas duas áreas estão relacionadas com seu avanço recente, principalmente a segunda. Ela faz uso da análise preditiva para tentar estabelecer a melhor decisão possível, o que casa perfeitamente com *Machine Learning*, uma vez que tenta prever a saída mais acurada através de milhares de entradas. Entretanto, para um modelo de *Machine Learning* ser bem-sucedido ele precisa de um grande e significativo volume de dados, o que casa perfeitamente com *Big Data*. Embora este termo sofra críticas por representar uma glamourização do termo “Estatística”, o fato é que Estatística e Computação caminham lado a lado nesta área.

Neste capítulo, serão abordados os assuntos presentes na ciência de dados: Aprendizado de Máquina (*Machine Learning*) e Reconhecimento de Padrões, que envolve principalmente o Processamento de Imagens. Após tais abordagens, será abordado como tais conceitos se encaixariam no contexto de Redes de Computadores, mais precisamente nos temas abordados no capítulo anterior.

3.1 APRENDIZADO DE MÁQUINA (MACHINE LEARNING)

Copeland (2016) define o aprendizado de máquina (*machine learning*, em inglês) como “a prática de usar algoritmos para coletar dados, aprender com eles, e então fazer uma

determinação ou predição sobre alguma coisa no mundo. Então em vez de implementar as rotinas de software manualmente, com um gama específica de instruções para completar uma tarefa em particular, a máquina é ‘treinada’ usando uma quantidade grande de dados e algoritmos que dão a ela a habilidade de aprender como executar a tarefa”. Com isso, o computador consegue a habilidade de realizar determinado cálculo ou tarefa sem que necessite de programação adicional ou interferência humana para isso.

O *machine learning* é fortemente relacionado com a Estatística, uma vez que seus métodos e parte de seus algoritmos, como regressões, tiveram como base modelos estatísticos e a análise de seus dados. As tarefas de aprendizado podem ser classificadas em três categorias básicas [17][18]:

- **Aprendizado supervisionado:** O treinamento é realizado por meio de exemplos rotulados, como uma entrada na qual a saída desejada é conhecida. Através de métodos como classificação, regressão e *gradient boosting*, o aprendizado supervisionado utiliza padrões para prever os valores de rótulos em dados não-rotulados adicionais. O aprendizado supervisionado é comumente empregado em aplicações nas quais dados históricos preveem eventos futuros prováveis.
- **Aprendizado não-supervisionado:** É utilizado em dados que não possuem rótulos históricos. A “resposta certa” não é informada ao sistema, o algoritmo deve descobrir o que está sendo mostrado. O objetivo é explorar os dados e encontrar alguma estrutura dentro deles. Técnicas populares incluem mapas auto-organizáveis, mapeamento por proximidade, agrupamento *k-means* e decomposição em valores singulares. Esses algoritmos também são utilizados para segmentar tópicos de texto, recomendar itens e identificar pontos discrepantes nos dados.
- **Aprendizado por reforço:** O algoritmo descobre através de testes do tipo “tentativa e erro” quais ações rendem as maiores recompensas. Este tipo de aprendizado possui três componentes principais: o agente (o aprendiz ou tomador de decisão), o ambiente (tudo com que o agente interage) e ações (o que o agente pode fazer). O objetivo é que o agente escolha ações que maximizem a recompensa esperada em um período de tempo determinado. O agente atingirá o objetivo muito mais rápido se seguir uma boa política, então o foco do aprendizado por reforço é descobrir a melhor política.

Nos últimos anos, o termo *machine learning* se tornou muito mais evidente com a possibilidade da implementação do *deep learning*, que é uma técnica que utiliza Redes

Neurais Artificiais para atingir seus resultados. Redes Neurais Artificiais são modelos computacionais inspirados no sistema nervoso do cérebro, onde temos neurônios divididos em camadas e conectados entre si, podendo ser abstraído conforme ilustração na Figura 9. Dependendo da tarefa a ser realizada, cada neurônio atribui um peso para os dados que entram e a saída final é determinada pelo total desses pesos (Copeland, 2016). As redes neurais utilizadas em *deep learning* possuem, ao menos, duas camadas de neurônios entre a camada que recebe os dados de entrada e a camada final que faz o tratamento final dos dados de saída.

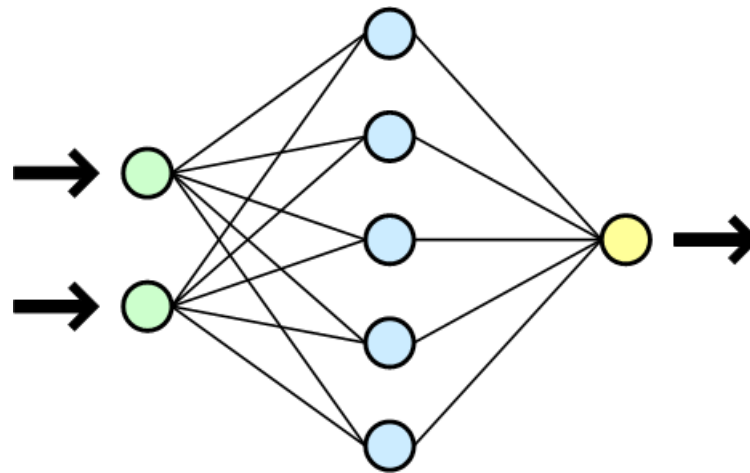


Figura 9. Diagrama simplificado de uma rede neural.

Com a capacidade de processamento atual, o treino de uma tarefa passou a ser cada vez mais viável, uma vez que métodos de *machine learning*, especialmente envolvendo redes neurais, são muito custosos computacionalmente. E sua viabilidade é acompanhada de efetividade: Como exemplo, reconhecimento de imagens por máquinas treinadas através de *deep learning* em alguns cenários possuem uma taxa de acerto maior que a de humanos (Copeland, 2016).

3.2 RECONHECIMENTO DE PADRÕES

O Reconhecimento de Padrões, segundo Theodoridis e Koutombras (1999), lida com a classificação automática de um determinado objeto em um de várias categorias (classes) diferentes. O objetivo dessa área é o desenvolvimento de um sistema de classificação automática, que será usado para a classificação de objetos relevantes para a aplicação em questão. Exemplos de aplicações onde pode-se aplicar esses classificadores são o reconhecimento de fala, reconhecimento de escrita e aplicações envolvendo Processamento de Imagens como reconhecimento de faces e códigos de barras.

Theodoridis e Koutombras (1999) também definem as seguintes etapas no *design* de um sistema de classificação:

- **Geração de *features*:** Esta etapa lida com a geração das características (*features*) que serão usadas para representar um objeto. Elas devem ser escolhidas para que seus valores variem significativamente entre as classes. Sua escolha depende da aplicação e geralmente são escolhidos em cooperação com especialistas no campo de aplicação.
- **Seleção de *features*:** Esta etapa lida com a seleção de tais *features*, dado um grande número de *features* geradas na etapa anterior, que são mais representativas do problema. Ou seja, são ricas em informações sobre a tarefa de classificação. Basicamente, seus valores devem variar muito para as diferentes classes. Esta é uma tarefa importante. Se os recursos selecionados forem ruins em informações relacionadas à classificação, o resultado geral do classificador será ruim.
- **Projeto do classificador:** Esta etapa pode ser considerada como o coração do sistema de classificação. O *design* do classificador geralmente é realizado através da otimização de um critério favorável. A tarefa de um classificador é dividir o espaço de *features* em regiões, cada uma das quais corresponde a uma classe. Assim, para um determinado valor, o classificador identifica a região no espaço de *features* a que pertence e atribui o valor à classe correspondente.
- **Avaliação do sistema:** A etapa final consiste no teste e avaliação de desempenho do sistema, que é a estimativa da probabilidade de erro de classificação. Essa também é uma tarefa importante, pois na maioria dos casos, essa estimativa é baseada em um número limitado de dados de teste. Se a probabilidade de erro chegar a ser maior que um limite preestabelecido, é capaz que o projetista precise reprojeter alguns ou todos os estágios anteriores.

Métodos utilizados no *machine learning*, como o *k-means*, também são utilizados no Reconhecimento de Padrões. Como sua base também deriva da estatística, tais áreas podem co-existir mutuamente. A grande diferença é que, enquanto o *machine learning* tenta prever certos dados através de comportamento indutivo de dados passados, o Reconhecimento de Padrões tenta apenas classificar dados passados através de comportamento indutivo ou dedutivo.

3.3 APLICAÇÕES NO CENÁRIO DE REDES DE COMPUTADORES

O uso de Ciência de Dados casa perfeitamente com o uso de *Fog computing*, podendo ser aplicado por exemplo em indústrias como a de saúde e a automobilística, onde o bombardeamento de dados dos sensores utilizados é aproveitado para obter um melhoramento contínuo dos modelos de classificação. Dadas as informações apresentadas no capítulo 2 e no capítulo 3, é possível traçar quais as etapas de um projeto aplicado a Ciência de Dados em uma arquitetura de *Fog-to-Cloud*, ilustradas na Figura 10.

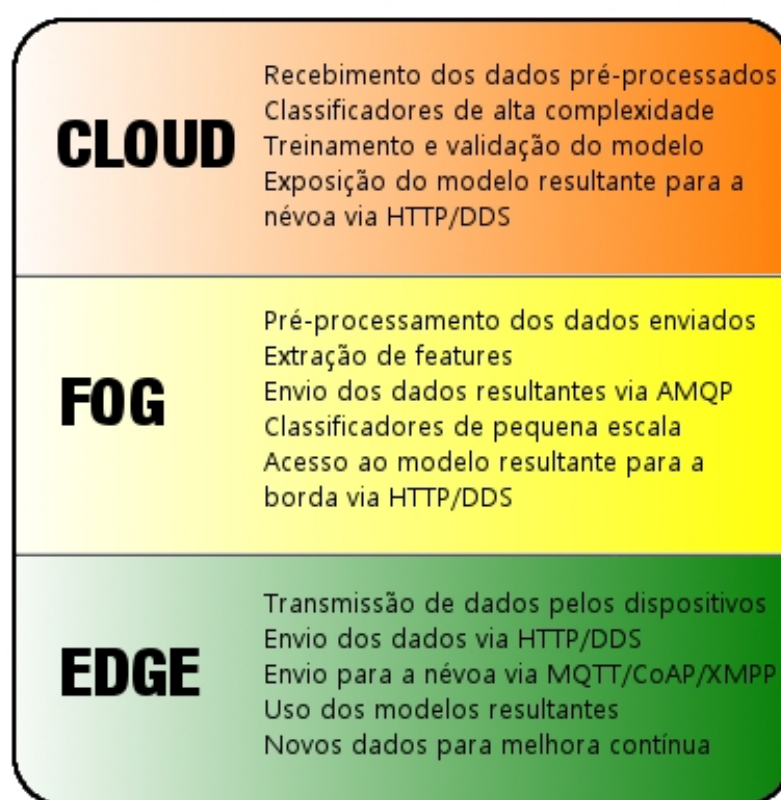


Figura 10. Etapas do Aprendizado de Máquina e do Reconhecimento de Padrões em uma arquitetura F2C.

Os dispositivos IoT, a princípio, ficariam encarregados apenas pela transmissão de dados pela rede, via HTTP ou DDS, e uma aplicação presente na borda da rede ficaria encarregada para que esses dados cheguem aos servidores presentes na névoa utilizando MQTT, CoAP ou XMPP. Uma vez que o treino no aprendizado de máquina é extremamente custoso e demorado, tal parte ficaria deslocada apenas para a nuvem, enquanto um pré-processamento dos dados poderia ser hospedada na névoa para poupar processamento da nuvem apenas para o treino, levando os dados processados para a nuvem utilizando AMQP. Após o treino ser finalizado, chamadas HTTP ou DDS podem ser encadeadas para levar o

modelo obtido aos dispositivos IoT para fazerem as predições e melhorarem o modelo continuamente.

No caso do reconhecimento de padrões, em qual camada o classificador ficaria dependeria de sua complexidade: Enquanto classificadores mais simples, como os de reconhecimento de código de barras, podem ser hospedados apenas na névoa sem grandes problemas, atividades como reconhecimento de escrita envolvem classificadores mais complexos por deus dados envolverem certas subjetividades e seriam mais adequados para serem hospedados na nuvem, podendo a névoa ser utilizada na extração de *features*.

4 TRABALHOS COLETADOS E COMPARAÇÕES

Embora seja possível traçar um desenho da arquitetura de um projeto de ciência de dados via *Cloud* e *Fog computing* usando apenas referências bibliográficas e um pouco de experiência prática, o desenho de onde cada função deste projeto se situa não passa apenas de mera suposição teórica. Para verificar a acurácia deste desenho e a aplicação prática desta teoria, foram coletados 10 trabalhos a respeito de diferentes aplicações no aprendizado de máquina e reconhecimento de padrões. Como apenas um deles é relativo ao reconhecimento de padrões propriamente dito e trabalhos relativos a aprendizado de máquina também podem ser inseridos nessa categoria dependendo do contexto da aplicação, a intenção deste levantamento não é comparar qual categoria é preferencial a outra, mas discutir a viabilidade destes projetos dentro de um cenário da rede.

Com exceção de um dos trabalhos que é datado de 2012, todos os outros trabalhos coletados foram publicados a partir de 2017, mostrando que a combinação é nova para a academia e há muito o que pesquisar a respeito do tema. Isso também pode explicar a grande diversidade de resultados deste levantamento, visto que cada situação levou a soluções completamente diferentes da imaginada no capítulo anterior, conforme mostrado em tabela comparativa localizada abaixo.

Também é possível reparar nas diferentes abordagens entre os trabalhos: Tuli et al. (2019), Lu, L. et al. (2018), Wu et al. (2017) e Scirè et al. (2019) não mencionam muitos detalhes sobre a arquitetura da rede, focando mais no desenvolvimento do algoritmo. Por outro lado, Lavassani et al. (2018), Soyata et al. (2012) e Alturki et al. (2019) mostram detalhes como eficiência de pacotes, tempo de resposta e comparativos entre diferentes arquiteturas.

Enquanto a maioria dos dispositivos IoT era responsável apenas pelo envio de dados, Nikouei et al. (2018) e Lu, L. et al. (2018) usavam o modelo na borda e Lavassani et al. (2018) realizava um pré-processamento básico antes de ir para a névoa. Situação semelhante apareceu na camada da nuvem: Enquanto a maioria das aplicações usava a nuvem para treinamento do modelo, Yang et al. (2018) usa a nuvem para extração de *features* e envio

Tabela 1. Tabela comparativa entre os trabalhos coletados. (ML = Machine Learning. RP = Reconhecimento de Padrões)

Trabalho	Descrição	Funções da borda/IoT	Funções da névoa/cloudlet	Funções da nuvem	Observações adicionais
Alturki et al. [22]	Comparativo de desempenho entre as arquiteturas de rede em diferentes algoritmos de ML	Transmissão e envio dos dados para a névoa	Diferentes funções, dependendo do experimento	Diferentes funções, dependendo do experimento	Conclui que uma arquitetura híbrida é mais eficiente que uma apenas baseada em <i>cloud computing</i> ou em <i>fog computing</i> , propondo funções semelhantes às mostradas nesta monografia
Nikouei et al. [23]	Uso de ML/RP e <i>Fog computing</i> para câmeras de segurança	Transmissão e envio dos dados para a névoa, uso do modelo, envio de dados para a névoa para melhora contínua	Extração de <i>features</i> , pré-processamento dos dados, envio dos dados para a nuvem ou treinamento e validação do modelo	Treinamento e validação do modelo, exposição do modelo para a névoa	Apenas teoriza onde cada etapa poderia se encaixar
Tuli et al. [24]	Uso de ML/RP e <i>Fog computing</i> para câmeras de segurança	Transmissão e envio dos dados para a névoa	Envio dos dados para a nuvem, treinamento e validação do modelo	Treinamento e validação do modelo	Não menciona pré-processamento dos dados e extração de <i>features</i>
Soyata et al. [25]	Uso de RP e <i>cloudlets</i> para detecção de rostos	Transmissão e envio dos dados para a névoa	Pré-processamento dos dados, envio dos dados para a nuvem, exposição do modelo	Uso do modelo, exposição do modelo	Tempo de resposta entre 100 ms e 1s dependendo do número de servidores na nuvem, do uso de <i>cloudlets</i> e do algoritmo utilizado
Lu, L. et al. [26]	Uso de ML/RP e <i>Fog computing</i> para reconhecimento de músicas	Uso do modelo	Exposição do modelo para a nuvem	Exposição do modelo para a borda	Não especifica se usa os nós da rede para treinamento ou se ele é realizado a parte do sistema
Wu et al. [27]	Uso de ML e <i>Fog computing</i> para automatização de máquinas industriais	Transmissão e envio dos dados para a névoa	Transmissão e envio dos dados para a nuvem	Treinamento e validação do modelo, exposição do modelo para a névoa	Não menciona pré-processamento dos dados e extração de <i>features</i>
Lavassani et al. [28]	Uso de ML e <i>Fog computing</i> para aumento de eficiência energética e de transmissão de dados para uso industrial	Transmissão e envio dos dados para a névoa, pré-processamento dos dados	Treinamento e validação do modelo, exposição do modelo, envio dos dados para a nuvem	Armazenamento dos dados, exposição do modelo	Usa MQTT para transmitir dados da névoa para nuvem, e USB (Serial) para transmitir dados da névoa para borda. Redução de 98% de pacotes enviados. Névoa utilizada para filtrar o
Scirè et al. [29]	Uso de ML/RP e <i>Fog computing</i> para detecção de arritmias cardíacas	Transmissão e envio dos dados para a névoa	Pré-processamento dos dados, envio dos dados para a nuvem	Treinamento e validação do modelo	bombardamento de dados dos dispositivos IoT. Não especifica a exposição do modelo
Lu, J. et al. [30]	Uso de ML e <i>Fog computing</i> no desenvolvimento de redes cognitivas via ondas de rádio	Transmissão e envio dos dados para a névoa	Extração de <i>features</i> , pré-processamento dos dados, envio dos dados para a nuvem, uso do modelo	Treinamento e validação do modelo, exposição do modelo para a névoa	Pré-processamento dos dados feito para detecção de anomalias
Yang et al. [31]	Uso de ML e <i>Fog computing</i> em privacidade na agregação dos dados de um dispositivo IoT	Transmissão e envio dos dados para a névoa	Pré-processamento dos dados, envio dos dados para a nuvem, treinamento, validação e uso do modelo	Extração de <i>features</i> , envio dos dados para a névoa para melhora contínua	Usa o processamento da nuvem para agregação de dados, que por sua vez melhora o modelo localizado na névoa

de dados para a névoa realizar o treinamento, uma vez que em sua solução o processamento da nuvem é melhor aproveitado agregação de dados, enquanto a névoa realiza um treinamento de uma rede neural mais leve para não sobrecarregá-la. A camada relativa a névoa foi a que teve mais diferentes abordagens em relação a sua premissa original, onde o poder de processamento dos servidores presentes na névoa, embora inferiores a nuvem, e sua proximidade com a borda fazem com que ela tenha versatilidade e pode compartilhar funções que em teoria seriam de suas camadas adjacentes.

Enquanto os trabalhos de Lu, J. et al. (2017), Alturki et al. (2019) (em apenas uma de suas abordagens), Lu, L. et al. (2018) e Nikouei et al. (2018) se aproximam mais da premissa abordada anteriormente, os trabalhos de Yang et al. (2018) e Lavassani et al. (2018) são os que mais diferem, mostrando que cada aplicação precisa ser estudada antes de seguir uma receita. Lavassani et al. (2018) merece destaque nesse ponto, pois mostra uma premissa totalmente diferente devido a seu foco: Enquanto o foco da premissa apresentada era para aplicar o desenvolvimento de um sistema baseado em ciência de dados a uma arquitetura baseada em *Cloud* e *Fog computing*, o foco de Lavassani et al. (2018) foi a eficiência energética e de envio de pacotes, fazendo não apenas mudar as funções das camadas da rede, mas também a escolha de dispositivos e até mesmo protocolos utilizados, preferindo protocolos mais leves como o MQTT para fazer a comunicação com a nuvem.

5 CONCLUSÃO

Cloud computing é um conceito que já possui fundamento sólido na sociedade atual e *Fog computing* ainda está se consolidando. Com a popularização dos dispositivos compatíveis com IoT, seu nome estará ainda mais em evidência no cenário de tecnologia. Quanto a ciência de dados, o constante avanço na capacidade de processamento e surgimento de novas arquiteturas para expandir ainda mais faz com que tal área seja uma parte fundamental na tomada de decisões de qualquer companhia.

O levantamento apresentado mostrou que ainda não há abordagens específicas para as diferentes aplicações, sendo necessário mais estudo sobre o tema e descoberta de diferentes padrões de arquitetura. Um sinal evidente deste fato foi a revisão bibliográfica inicial, que apresentou características que levou a um desenho inicial diferente de todos os trabalhos apresentados. Entretanto, o mesmo também mostrou que uma arquitetura *Fog-To-Cloud* em uma aplicação voltada a ciência de dados é perfeitamente viável, e até mais eficiente que uma arquitetura apenas baseada em *Cloud computing* ou uma arquitetura apenas baseada em qualquer outro tipo de *Edge computing*.

Como trabalhos futuros, é possível realizar experimentos práticos com o uso de um sistema baseado em ciência de dados em uma arquitetura no modelo teórico proposto no capítulo 3, para verificar sua viabilidade, comparações de performance com outras arquiteturas e definir um padrão que se encaixe em diferentes aplicações.

6 BIBLIOGRAFIA

- [1] REGALADO, Antonio, Who Coined 'Cloud Computing'?, 31 Out. 2011, presente em: <<https://www.technologyreview.com/s/425970/who-coined-cloud-computing/>> Acesso em: 30 Out. 2019
- [2] Cloud computing vs fog computing vs. edge computing na era da internet das coisas industrial, 19 Set. 2019, presente em: <<http://www.ccg.pt/cloud-computing-vs-fog-computing-vs-edge-computing-na-internet-das-coisas-industrial/>> Acesso em: 30 Out. 2019
- [3] Wikipedia, Cloud computing, presente em: <https://en.wikipedia.org/wiki/Cloud_computing> Acesso em: 30 Out. 2019
- [4] Wikipedia, Amazon Web Services, presente em: <https://en.wikipedia.org/wiki/Amazon_Web_Services> Acesso em: 30 Out. 2019
- [5] MELL, Peter, GRANCE, Timothy, The NIST Definition of Cloud Computing (Technical report). National Institute of Standards and Technology: U.S. Department of Commerce, 19 Oct. 2018, págs 2-3, presente em: <<https://medium.com/@thinkwik/how-edge-and-fog-computing-are-taking-over-traditional-cloud-computing-b26b7276f1ce>> Acesso em: 30 Out. 2019
- [6] How Edge and Fog Computing are taking over Traditional Cloud Computing, 19 Oct. 2018, presente em: <<https://medium.com/@thinkwik/how-edge-and-fog-computing-are-taking-over-traditional-cloud-computing-b26b7276f1ce>> Acesso em: 30 Out. 2019
- [7] OLIVEIRA, Bruno, Fog #01 – Introdução a computação em Névoa, 1 Jun. 2019, presente em: <https://medium.com/@bruno_live/fog-01-introdu%C3%A7%C3%A3o-a-computa%C3%A7%C3%A3o-em-n%C3%A9voa-d4303d5f90b9> Acesso em: 31 Out. 2019
- [8] DILLEY, John et al., Globally Distributed Content Delivery, IEEE Internet Computing, Volume 6, Edição 5, Nov. 2002, presente em: <https://people.cs.umass.edu/~ramesh/Site/PUBLICATIONS_files/DMPPSW02.pdf> Acesso em: 31 Out. 2019
- [9] Edge Computing, Fog Computing, Cloudlets, Mobile Edge Computing: Um novo modelo de Computação, 26 Jan. 2019, presente em: <<http://blog.ccna.com.br/2019/01/26/edge->

[computing-fog-computing-cloudlets-mobile-edge-computing-um-novo-modelo-de-computacao/](#)> Acesso em: 31 Out. 2019

[10] OLIVEIRA, Bruno, Fog #02 – Computação na borda da rede, 1 Jun. 2019, presente em: <https://medium.com/@bruno_live/fog-02-computa%C3%A7%C3%A3o-na-borda-da-rede-83fca9970c44> Acesso em: 31 Out. 2019

[11] Wikipedia, OpenFog Consortium, presente em: <https://en.wikipedia.org/wiki/OpenFog_Consortium> Acesso em: 31 Out. 2019

[12] OLIVEIRA, Bruno, Fog #03 – Arquitetura de computação em névoa, 1 Jun. 2019, presente em: <https://medium.com/@bruno_live/fog-03-arquitetura-de-computa%C3%A7%C3%A3o-em-n%C3%A9voa-69b2d8bbf1f7> Acesso em: 31 Out. 2019

[13] OpenFog Consortium, Open Fog Reference Architecture for Fog Computing, 8 Fev. 2017, presente em: <https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf> Acesso em: 31 Out. 2019

[14] DIZDAREVIĆ, Jasenka et al., A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration, ACM Computing Surveys, Volume 1, Número 1, Fev. 2019, presente em: <<https://arxiv.org/pdf/1804.01747.pdf>> Acesso em: 15 Nov. 2019

[15] Wikipedia, Ciência de dados, presente em: <https://pt.wikipedia.org/wiki/Ci%C3%Aancia_de_dados> Acesso em: 30 Out. 2019

[16] COPELAND, Michael, What's the Difference Between Artificial Intelligence, Machine Learning and Deep Learning?, 29 Jul. 2016, presente em: <<https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>> Acesso em: 30 Out. 2019

[17] Wikipedia, Aprendizado de Máquina, presente em: <https://pt.wikipedia.org/wiki/Aprendizado_de_m%C3%A1quina> Acesso em: 30 Out. 2019

[18] SAS Insights, Machine learning: o que é e qual sua importância?, presente em: <https://www.sas.com/pt_br/insights/analytics/machine-learning.html> Acesso em: 30 Out. 2019

[19] THEODORIDIS, Sergios, KOUTROUMBAS, Konstantinos, Pattern Recognition and Neural Networks, ACAI 1999: Machine Learning and Its Applications, págs. 169-170, presente em: <https://link.springer.com/chapter/10.1007/3-540-44673-7_8>

- [20] Wikipedia, Reconhecimento de padrões, presente em: <https://pt.wikipedia.org/wiki/Reconhecimento_de_padr%C3%B5es> Acesso em: 31 Out. 2019
- [21] VALIN, Allan, Inteligência Artificial: reconhecimento de padrões, 29 Out. 2009, presente em: <<https://www.tecmundo.com.br/seguranca/3014-inteligencia-artificial-reconhecimento-de-padroes.htm>> Acesso em: 31 Out. 2019
- [22] ALTURKI, Badraddin et al., Smart Surveillance as an Edge Network Service: from Harr-Cascade, SVM to a Lightweight CNN, IEEE Transactions Sustainable Computing, Mar. 2019, presente em: <https://www.researchgate.net/publication/332089532_Exploring_the_Effectiveness_of_Service_Decomposition_in_Fog_Computing_Architecture_for_the_Internet_of_Things> Acesso em: 17 Nov. 2019
- [23] NIKOUEI, Seyed Yahya et al., Smart Surveillance as an Edge Network Service: from Harr-Cascade, SVM to a Lightweight CNN, 2018 IEEE 4th International Conference on Collaboration and Internet Computing, Out. 2018, presente em: <https://www.researchgate.net/publication/329070876_Smart_Surveillance_as_an_Edge_Network_Service_From_Harr-Cascade_SVM_to_a_Lightweight_CNN> Acesso em: 17 Nov. 2019
- [24] TULI, Shreshth et al., EdgeLens: Deep Learning based Object Detection in Integrated IoT, Fog and Cloud Computing Environments, Jun. 2019, presente em: <https://www.researchgate.net/publication/334049225_EdgeLens_Deep_Learning_based_Object_Detection_in_Integrated_IoT_Fog_and_Cloud_Computing_Environments> Acesso em: 17 Nov. 2019
- [25] SOYATA, Tolga et al., Cloud-vision: Realtime face recognition using a mobile-cloudletcloud acceleration architecture, Jan. 2012, presente em: <https://www.researchgate.net/publication/312437716_Cloud-vision_Realtime_face_recognition_using_a_mobile-cloudletcloud_acceleration_architecture> Acesso em: 17 Nov. 2019
- [26] LU, Lifei et al., Fog Computing Approach for Music Cognition System Based on Machine Learning Algorithm, IEEE Transactions On Computational Social Systems, Out. 2018, presente em:

<https://www.researchgate.net/publication/328207760_Fog_Computing_Approach_for_Music_Cognition_System_Based_on_Machine_Learning_Algorithm> Acesso em: 17 Nov. 2019

[27] WU, Dazhong et al., A fog-computing-based framework for process monitoring and prognosis in cyber-manufacturing, 28 Fev. 2017, presente em: <https://www.researchgate.net/publication/314133324_A_Fog_Computing-based_Framework_for_Process_Monitoring_and_Prognosis_in_Cyber-Manufacturing>

Acesso em: 17 Nov. 2019

[28] LAVASSANI, Mehrzad et al., Combining Fog Computing with Sensor Mote Machine Learning for Industrial IoT, Mai. 2018, presente em: <https://www.researchgate.net/publication/325122606_Combining_Fog_Computing_with_Sensor_Mote_Machine_Learning_for_Industrial_IoT> Acesso em: 17 Nov. 2019

[29] SCIRÈ, Alessandro et al., Fog-Computing-Based Heartbeat Detection and Arrhythmia Classification Using Machine Learning, 2 Fev. 2019, presente em: <https://www.researchgate.net/publication/330891603_Fog-Computing-Based_Heartbeat_Detection_and_Arrhythmia_Classification_Using_Machine_Learning>

Acesso em: 17 Nov. 2019

[30] LU, Jingyang et al., Machine Learning based Intelligent Cognitive Network using Fog Computing, Nov. 2017, presente em: <https://www.researchgate.net/publication/321347270_Machine_Learning_based_Intelligent_Cognitive_Network_using_Fog_Computing> Acesso em: 17 Nov. 2019

[31] YANG, Mengmeng et al., Machine Learning Differential Privacy with Multifunctional Aggregation in a Fog Computing Architecture, IEEE Access, Mar. 2018, presente em: <<https://www.tecmundo.com.br/seguranca/3014-inteligencia-artificial-reconhecimento-de-padroes.htm>> Acesso em: 17 Nov. 2019